devsu

The Cost of Standing Still

What Legacy Systems Are Really Costing US Financial Institutions in 2026

A Devsu publication for CTOs, VPs of Engineering, and CFOs at US banks, credit unions, insurers, and fintechs.

Edition 2026

Content

- **00 - Executive summary**
- **01 - The Five Hidden Costs of Legacy Systems**
  - Maintenance cost inflation — the run-the-bank tax
  - Opportunity cost of delayed feature delivery
  - Regulatory exposure cost
  - Talent cost — the COBOL retirement problem
  - Customer attrition cost
- **02 - The Modernization Priority Matrix**
  - Customer-facing channels — modernize first
  - Identity, fraud, and AML — modernize second
  - Integration and API layers — modernize third
  - Data infrastructure and analytics — modernize fourth
  - Core processing — modernize last, in pieces
- **03 - The Executive Decision Framework**
  - Three decisions executives keep deferring
  - Building the modernization business case in board-readable language
  - The false economy of "next year"
  - The cost-of-standing-still worksheet
- **04 - The Talent Reality**
  - Supply and demand: the data
  - Wage inflation and the budget consequences
  - Why waiting makes the talent problem worse
  - Four credible responses (with honest tradeoffs)

Content

- **05 - The Real Stories**
  - Institution A — top-five regional bank (Devsu engagement, 2024–2025)
  - Institution B — core transaction modernization (Devsu engagement, 2025)
  - Mid-size credit union (composite scenario)
  - Mid-market P&C insurance carrier (composite scenario)
  - Institution C — Series B+ consumer-credit fintech (composite, anchored to real engagement)
- **06 - The Path Forward**
  - The bounded pilot: what to pick
  - The 90-day proof-of-value approach
  - The questions executives should ask any vendor (including Devsu)
  - What the next 90 days should look like
  - Closing
- **Appendix A: Formula Index**
- **Appendix B: References**

Section 0

Executive Summary

Every financial institution running legacy systems is paying a recurring tax. Most do not know its size. The tax shows up across five line items — maintenance, delayed delivery, regulatory exposure, talent, and customer attrition — and it compounds. Banks already commit between 6% and 12% of revenue to IT, the highest share of any major industry, and 10% to 20% of every new project is consumed by tech debt before any new value is delivered [1]. For an institution running on architecture from before the iPhone existed, that share is not a budgeting problem. It is a strategy problem.

This ebook does three things. It quantifies the five hidden costs with current data from McKinsey, Gartner, Deloitte, IBM, the OCC, the NCUA, the BLS, Cornerstone Advisors, and Celent. It ranks the systems that should be modernized first by risk-adjusted return. And it offers an honest framework for executives weighing whether to start now, defer another year, or commit to a multi-year program.

The argument is not that every institution should rip and replace. The argument is that the cost of waiting another year is now measurable, and in most cases it exceeds the cost of starting a bounded modernization effort this quarter. Section 3 includes a worksheet to estimate that figure for your own institution.

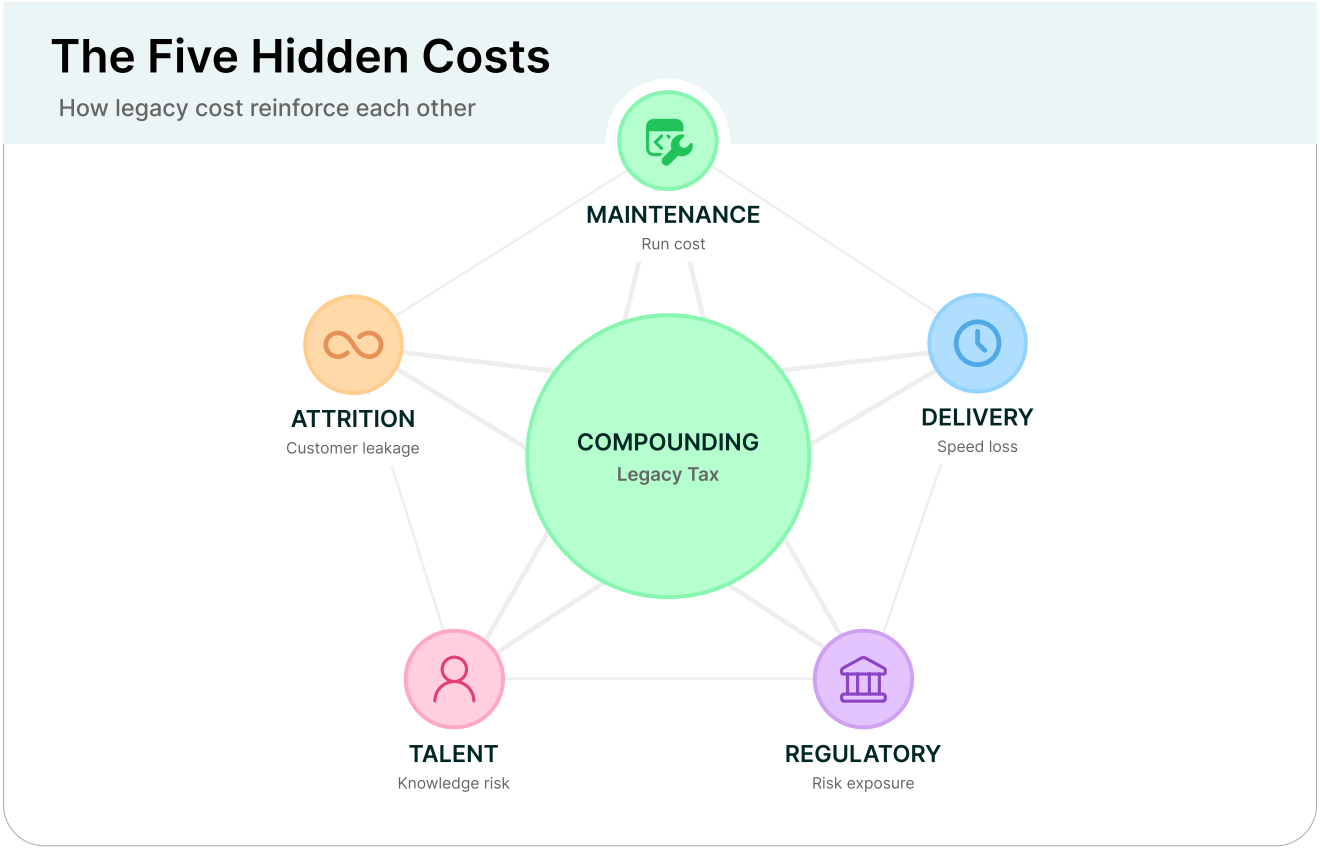
The asset is structured to be read in 30 minutes by a busy CTO and re-read in 60 by a CFO who wants to validate the numbers. Skim the sidebars and tables; read the formulas if you intend to build a board case.

Section 1

The Five Hidden Costs of Legacy Systems

The first thing legacy costs do is hide. They show up under different ledger lines, scattered across maintenance, vendor management, recruiting, compliance remediation, and lost revenue. When you add them up — which most institutions do not — the total tends to surprise the executive team.

This section walks through the five compounding costs every financial institution carries. For each, we give a definition, sourced data quantifying it, a formula you can apply to your own institution, and a real-world reference.



Maintenance cost inflation: the run-the-bank tax

The cost: Every dollar of IT budget locked into keeping existing systems alive is a dollar not available for new product, regulatory work, or AI. In banking, that share is unusually high.

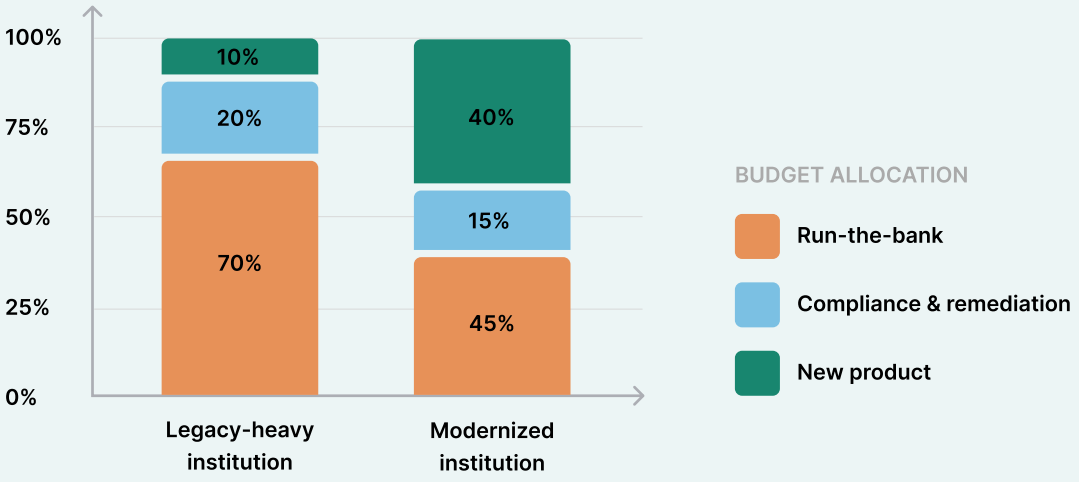
The data: Banking spends 6% to 12% of revenue on technology, the highest share of any major industry tracked by McKinsey, well ahead of telecom/media/tech at 3.75% to 5% [1]. Of that, McKinsey reports that institutions pay an additional 10% to 20% surcharge on every new project to absorb tech debt, and that roughly 30% of CIOs surveyed believe more than 20% of new-product budgets are lost to legacy drag.

A separate Deloitte banking technology survey estimates that institutions underestimate the true total cost of ownership of legacy core systems by 70% to 80%, with actual TCO running approximately 3.4× the originally budgeted figure once infrastructure, support, integration, compliance overhead, and innovation drag are included [2].

↗ View in Formula Index

Where the IT dollar goes

Budget allocation shift from legacy-heavy to modernized institutions



Source: McKinsey 2024; illustrative split based on public benchmarks

Reference point. McKinsey research on banking architecture finds that on modern modular, API-enabled stacks, new solutions reach production in three to four months. On legacy architecture the same work takes nine to 18 months [1]. That difference is itself a hidden cost, covered in 1.2.

Opportunity cost of delayed feature delivery

The cost: Every product launch that takes 12 months on legacy and 3 months on modern architecture costs the institution 9 months of revenue, customer acquisition, and competitive positioning. That gap rarely shows up on the IT budget. It shows up in flat fee income, declining checking-account share, and lost cross-sell.

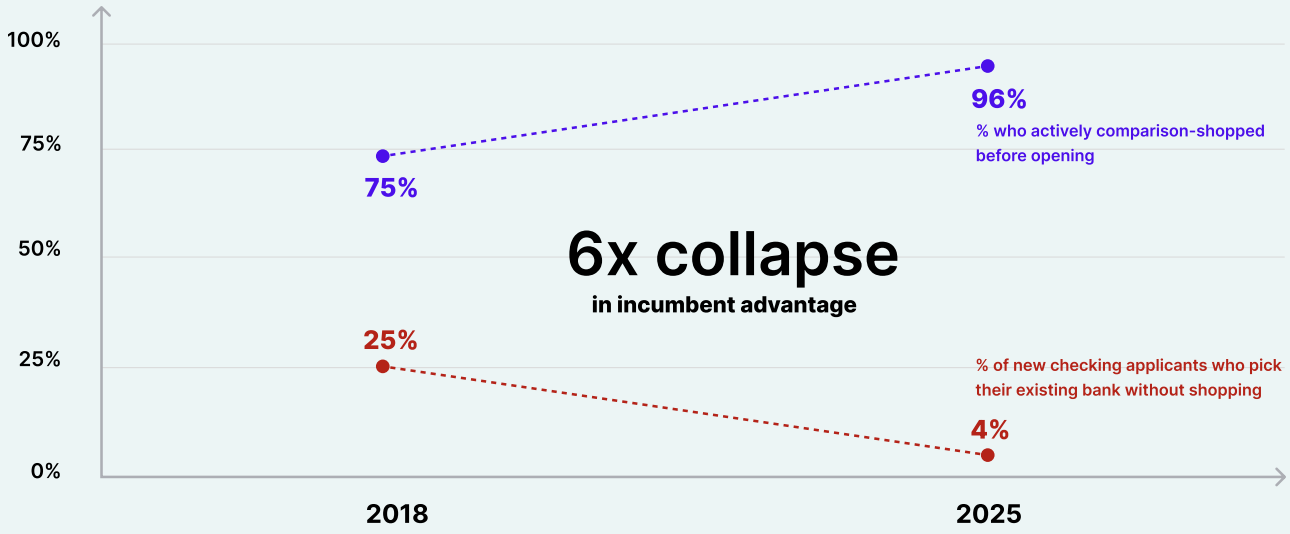
The data: McKinsey's 2025 Global Banking Annual Review finds that only 4% of new US checking-account applicants now choose their existing bank without shopping around, down from 25% in 2018 [3].

That is a six-fold collapse in incumbent advantage in seven years. The institutions winning new accounts are the ones shipping features fast enough to show up in comparison shopping.

McKinsey also estimates that if banks fail to capture the AI productivity opportunity, the central scenario is a $170 billion (9%) erosion of global banking profit pools, with early adopters gaining a roughly 4 percentage-point increase in Return on Tangible Equity [3].

↗ View in Formula Index

Where New Checking Accounts Go



Source: McKinsey Global Banking Annual Review 2025.

Reference point: Section 5 illustrates the inverse using an anonymized client engagement. A single bank-deposit transaction representing roughly 60% of the institution's daily cash transactions was modernized in seven days of focused development using AI-assisted refactoring. The same work, attempted by a small team without AI assistance and against undocumented .NET 2.0 code, had previously been deemed infeasible [4].

Regulatory exposure cost

The cost: Legacy systems generate compliance findings. Compliance findings generate consent orders. Consent orders generate remediation costs, restricted growth, and personal accountability for executives.

The data: Between June 2023 and June 2024, US federal banking regulators (the Federal Reserve, OCC, and FDIC) entered into more than 45 cease-and-desist orders with non-G-SIBs and at least 20 written or formal agreements; 12 of the C&Ds were full consent orders [5]. A non-trivial share of these specifically cite information technology controls. One regional bank's January 2024 consent order included a dedicated "Information Technology Control Program" article requiring board-approved IT risk governance, IT risk assessment, and data-integrity standards [6]. A separate major bank's February 2024 cease-and-desist also cited operational, compliance, and IT risk-management deficiencies [7].

The OCC has stated publicly that it is concerned about banks' reliance on legacy systems and is using both consent orders and "matters requiring attention" to push modernization.

The cost of inaction extends beyond enforcement. The 2024 IBM/Ponemon Cost of a Data Breach Report puts the average financial-services breach at $6.08 million in 2024 — 22% above the global average [8]. The 2025 edition reports the global average fell 9% but US breach costs hit a record $10.22 million, with financial services at $5.56 million on average [9].

↗ View in Formula Index

Reference point: When consent orders cite IT controls, the remediation work is typically a 12-to-24-month program with quarterly board reporting. The remediation is, in practice, a forced modernization done under regulator supervision and time pressure — usually at higher cost and with worse outcomes than a voluntary program would have produced.

Selected 2024 federal enforcement actions citing IT or operational controls

Institution profile	Action	IT-related findings cited
Mid-size community bank (mid-Atlantic)	Consent order + civil money penalty (Oct 2024)	IT controls, BSA/AML, capital, liquidity
Large California-based commercial bank	C&D + CMP + GLBA Agreement (Feb 2024)	Operational, compliance, investment-mgmt risk; heightened-standards noncompliance
Top-10 US bank (national)	C&D + CMP (Oct 2024)	BSA/AML transaction monitoring; "long-standing systemic breakdowns"
Global systemically important bank	$250M civil money penalty (Mar 2024)	Trade surveillance / oversight gaps

Source: OCC enforcement release archive, 2024; SEC 8-K filings of named banks. See Appendix B, refs [5]–[7].

Talent cost — the COBOL retirement problem

The cost: The engineers who built and maintain core banking systems are aging out. Their replacements are not being trained. Wages for the remaining specialists are inflating. And institutions are increasingly hostage to a single individual who happens to remember how a critical batch process works.

The data: Industry estimates put the in-production COBOL codebase at roughly 220 billion lines globally [10] (some 2022 estimates put the figure as high as 800 billion). COBOL still underpins roughly 43% of US banking systems, 95% of ATM transactions, and over 80% of in-person credit card transactions [11].

The talent problem is now structural. IBM and Fujitsu have separately reported that the average COBOL programmer is around 58 years old and roughly 10% retire each year [12]. Multiple labor-market analyses estimate that fewer than 2,000 COBOL programmers graduated worldwide in 2024 [13].

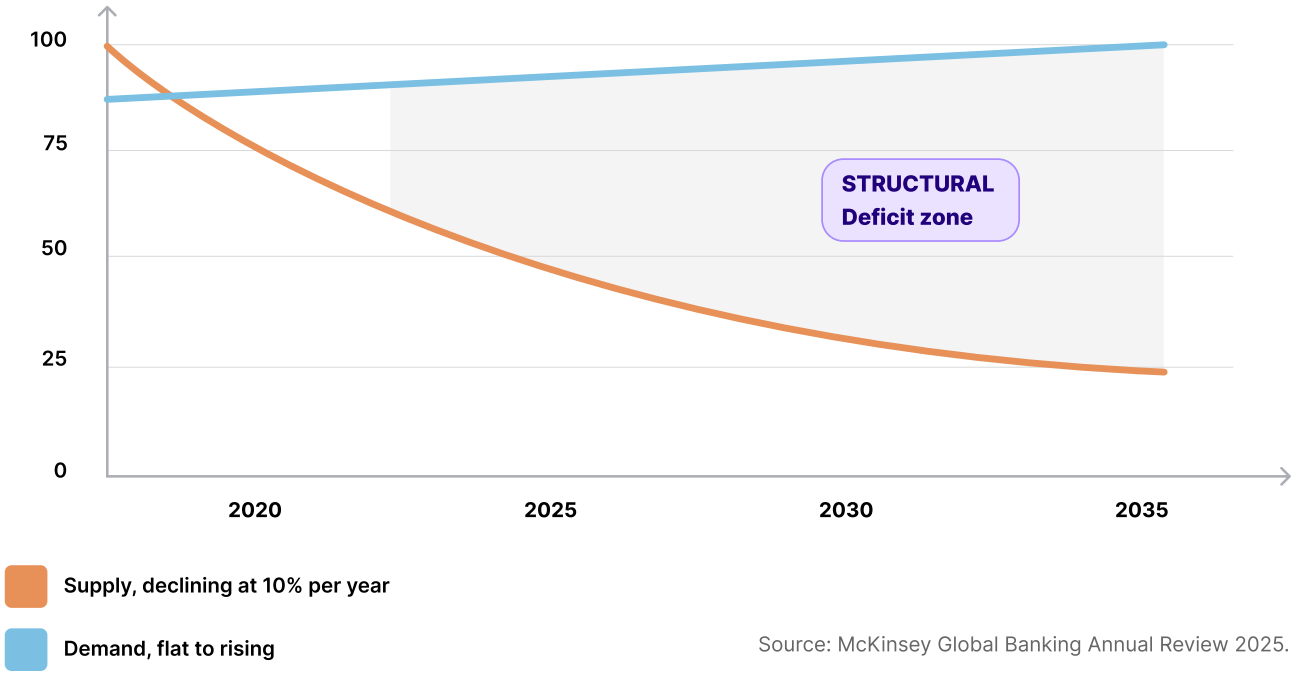
Salary data from public job-board aggregators put COBOL maintenance roles in the $86K–$121K range and modernization-consulting roles up to $150K, with persistent shortages projected through 2035 [13].

↗ View in Formula Index

Reference point: A frequently-told story in modernization circles: a major US retailer discovered that its 2,500 store locations all ran on COBOL programs totaling approximately 20 million lines of code, maintained by two engineers — one of whom was retiring [14]. The institutions that survive this transition are not the ones that find one more retired COBOL programmer. They are the ones that built tooling — including AI-augmented refactoring — to let modern engineers work productively on the legacy stack. Section 4 covers this in depth.

The COBOL Talent Gap

Specialist supply declines while institutional demand persist



Source: McKinsey Global Banking Annual Review 2025.

Customer attrition cost

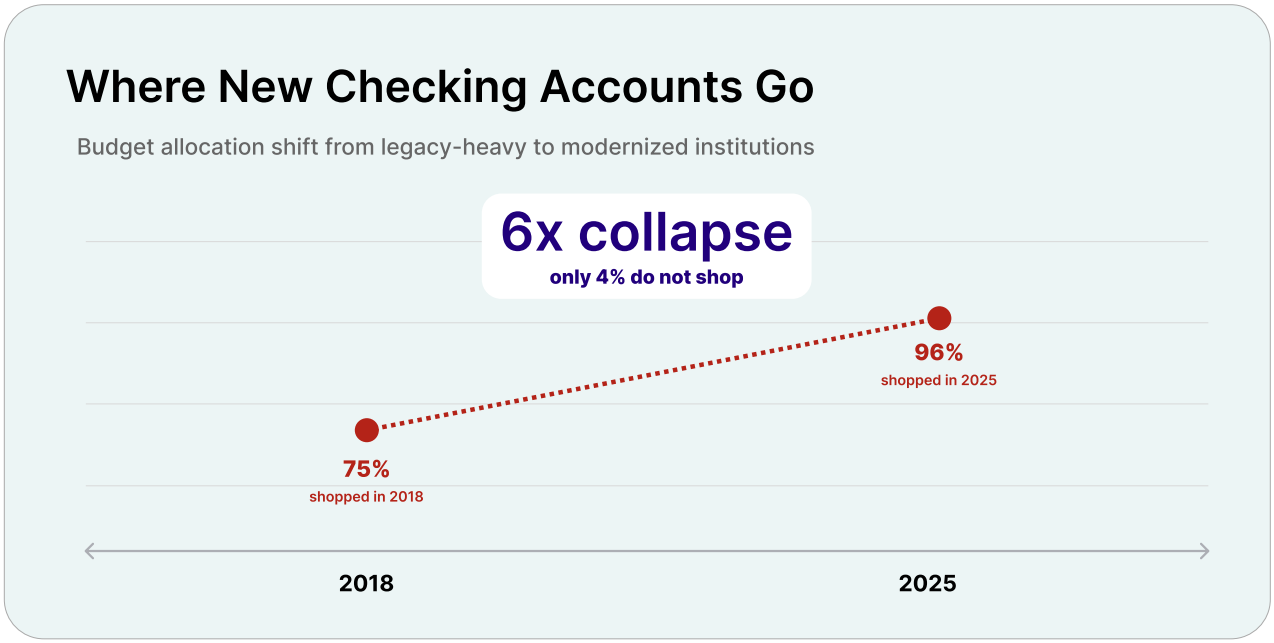
The cost: Customers who experience friction in digital channels do not file complaints. They open an account elsewhere. Most institutions discover the cost of their digital experience only in the deposit-attrition numbers two quarters later.

The data: The collapse of incumbent advantage is the single most striking finding in McKinsey's 2025 review: **only 4% of new US checking-account applicants now choose their existing bank without shopping, against 25% in 2018** [3].

The Cornerstone Advisors / Alkami 2025 Digital Banking Performance Metrics Report — the sixth annual edition, drawing on a sample of banks and credit unions averaging $4.7B in assets — found that average digital spending fell roughly 30% in 2024 after three years of double-digit growth, and warns that community banks and credit unions are "drowning in a sea of sameness" [15].

The implication is uncomfortable: the institutions that out-spent on digital are not differentiated; the institutions that under-invested are exposed.

Industry retention research consistently finds that the cost to acquire a new banking customer in the US is roughly 5× the cost to retain an existing one (~$390 vs. ~$75 per customer in commonly-cited industry estimates), and that a 5% lift in retention can produce 25% to 95% profit improvement depending on product [16].



Reference point: The same Cornerstone report observes that top banks' mobile app ratings are now consistently above 4.5/5.0 [15]. For a community institution running a 3.8-rated app, every download is a head-start surrendered to a megabank or fintech that the customer can install in 90 seconds.

Putting the five together.

Each of these costs is recoverable, but only one is recoverable on its own. Maintenance costs go down when systems are modernized. Delivery speed goes up. Regulatory exposure narrows. Talent risk diversifies. Attrition slows. They move together because they share a single underlying cause.

The next section answers the obvious follow-up question: if you can only modernize one thing this year, what is it?

Section 2

The Modernization Priority Matrix

If maintenance budgets were unconstrained and engineering capacity infinite, every legacy system would be on a roadmap. They are not. The question for any institution is not whether to modernize, but which order. This section ranks the five system layers by risk-adjusted three-year ROI and adapts the priority for banks under $50B AUM, credit unions, insurance carriers, and Series B+ fintechs.

The ranking below is opinionated. Reasonable executives will disagree on the order, particularly between core processing and data infrastructure. We have placed customer-facing channels above core processing intentionally: for most institutions, the channel layer is where attrition actually happens, and channel modernization can be done independently of a multi-year core replacement.

Modernization priority matrix

Highest priority fades from Devsu green to gray as time-to-value lengthens and dependency risk rises.

Rank	System layer	3-year cost of inaction (mid-size, $5–20B)	Min. viable modernization	Time to value
1	Customer-facing channels (mobile, web, contact center)	**$25M–$80M** in attrition + lost acquisition	New mobile/web shell over existing core via API gateway; 4–6 month effort	90–180 days
2	Identity, fraud, AML systems	**$5M–$50M** (single enforcement event) + breach exposure	Replace rules-only fraud engine with ML-based scoring; modern KYC vendor integration	120–240 days
3	Integration & API layers	**$10M–$30M** (delivery drag across all other layers)	API gateway + event bus; document and standardize top 20 internal APIs	90–270 days
4	Data infrastructure & analytics	**$15M–$60M** (decision lag + AI use-case backlog)	Modern data lake + governed product layer; one high-value use case shipped	180–360 days
5	Core processing	**$20M–$200M** (full TCO carry + concentration risk)	Strangler-pattern migration of one critical transaction; no big-bang	12–36 months

Sources: McKinsey 2024 banking technology research; Gartner banking IT-spend forecasts; IBM 2024–2025 breach cost data; aggregated Cornerstone benchmarking. Cost ranges are illustrative for institutions in the $5–20B asset range and should be adjusted to your specific footprint. See Appendix B, refs [1], [8], [9], [15].

Customer-facing channels — modernize first

Who feels the pain. The CMO, the head of retail, the CFO watching primary-deposit attrition. The CTO usually feels it last, because channel issues surface as customer complaints rather than infrastructure incidents.

Three-year cost of inaction. For a $5B-to-$20B-asset institution, the realistic three-year carry cost of an outdated digital channel runs **$25M to $80M** in some combination of net deposit attrition, foregone customer acquisition, and elevated marketing spend to overcome poor app ratings.

Minimum viable modernization. A new mobile and web shell built against a clean API gateway, sitting in front of the existing core. This is achievable in four to six months without a core replacement. The Institution C consumer-credit case in Section 5 shows the pattern: 144 hours of engineering per release cycle reduced to 40 hours, with a 30% lift in NPS, by modernizing the workflow layer rather than the underlying core [4].

Sub-vertical adaptations

Banks under $50B AUM

The single highest-payoff move is a credible mobile experience matched to the megabanks. Aim for 4.5 / 5.0 app store ratings as the minimum bar. Below that, you are paying acquisition costs without acquiring.

Credit unions

Members are stickier than bank customers, but the gap is closing. The 2025 Cornerstone report shows credit unions face the same "sea of sameness" as community banks [15]. Differentiation now means feature gaps closed (subscription management, real-time funds transfer, branch-equivalent self-service) rather than novel features.

Insurance carriers

The channel modernization argument is weaker — most insurance customer interactions are mediated by agents — but the agent-facing portal is now the channel. Agents who cannot get a quote in 90 seconds will place business with the carrier whose portal works.

Fintechs

Channel is the product. The question for a Series B+ fintech is whether the channel can scale 10× without a rewrite. Most cannot; the question is whether to refactor now or after a regulatory finding forces it.

Identity, fraud, and AML — modernize second

Who feels the pain: The Chief Risk Officer, the BSA officer, the General Counsel. And, increasingly, the CEO — one major bank's October 2024 consent order was specifically grounded in "long-standing systemic breakdowns" in transaction monitoring [17].

Three-year cost of inaction: Single-event exposure dominates. A median financial-services data breach now costs $5.56 million globally in 2025, and US-domiciled institutions average $10.22 million [9]. A BSA/AML enforcement event, by comparison, can run $25M to $250M+ in civil money penalty alone, before remediation. The OCC assessed a $250M CMP against a global systemically important bank in March 2024 [18].

Minimum viable modernization: Replace rules-only fraud and AML engines with machine-learning-based scoring; integrate a modern KYC/CIP vendor; consolidate identity into a single source of truth. This is typically 120-to-240 days of focused work and can be done independently of core modernization.

Sub-vertical adaptations

Banks

Federal banking regulators have moved decisively on AML modernization. Expect this to be the most-cited area in supervisory letters through 2027.

Credit unions

NCUA's 2025 supervisory priorities continue to emphasize information security and cybersecurity governance [19]. For credit unions under $1B in assets, modern fraud tooling is now available as packaged SaaS, build vs. buy strongly favors buy.

Insurance carriers

Fraud modernization is primarily a claims-side problem. The wins here come from ML-based claims triage rather than identity-side controls.

Fintechs

Identity and fraud are existential. A fintech with weak fraud controls will not survive its first major attack, and the major attack is statistically inevitable above a certain user base.

Integration and API layers — modernize third

Who feels the pain. Every engineering team. Integration debt is the cost that quietly inflates every other cost.

Three-year cost of inaction. Roughly $10M to $30M for a mid-size institution, mostly recoverable as delivery-speed gains across other modernization layers. McKinsey's finding that modern API-enabled architectures deliver in three to four months versus nine to 18 on legacy is largely an integration-layer story [1].

Minimum viable modernization. API gateway plus event bus; document and standardize the top 20 internal APIs that every other system depends on; deprecate point-to-point integrations on a schedule.

This is dull work. It is also the single highest-return technical investment most institutions can make.

Sub-vertical adaptations. Largely consistent across verticals, with one nuance: insurance carriers with multiple policy administration systems acquired through M&A face a more severe version of this problem than de-novo banks or fintechs. A typical large carrier has four-to-seven core PAS instances, often poorly integrated; the integration layer is the only place this gets solved short of a multi-year PAS consolidation.

Data infrastructure and analytics — modernize fourth

Who feels the pain. The CFO, the Chief Data Officer (where one exists), and any product manager trying to ship a feature that requires customer-level data.

Three-year cost of inaction. **$15M to $60M** for mid-size institution, including decision-lag costs, AI use-case backlog, and regulatory data-quality remediation. McKinsey reports that establishing a modern data and AI platform reduces time-to-insight for new analytics use cases from nine-to-12 months to two-to-three months [1].

Minimum viable modernization. A modern data lake or lakehouse, a governed data product layer, and one high-value use case shipped from raw data through to production decisioning within nine months. Resist the urge to re-platform all data before shipping anything; ship one use case to prove the pattern.

Sub-vertical adaptations

Banks and credit unions

Cornerstone's 2025 What's Going On in Banking survey found that only 6% of banks and 11% of credit unions rated their data strategy as "very effective," and roughly half of bank leaders said their use of data to enhance customer experience is "not effective" [20]. The bar is low; modest investment moves you out of the bottom quartile.

Insurance carriers

Data infrastructure is closer to a top-three priority for insurers than for banks, because actuarial and pricing capabilities depend on it. Celent's 2024 P&C Insurance IT Pressures & Priorities study found 91% of carriers cite growth/distribution as a significant or moderate priority [21] — most growth strategies are data-dependent.

Fintechs

Data-native by design, but accumulating warehouse-vs-application coupling debt as they scale.

Core processing — modernize last, in pieces

Who feels the pain. Eventually, everyone. Core systems are the slowest-moving layer in any financial institution and the highest-risk to touch.

Three-year cost of inaction. **$20M to $200M+** for a mid-size institution, including full TCO carry of the legacy core, concentration risk on retiring talent, and forgone optionality on every other layer that depends on the core. The high end of the range applies to institutions running mainframe-based COBOL cores with double-digit-year-old custom modifications.

Minimum viable modernization. Strangler-pattern migration of one critical transaction or product line, with explicit no-rip-and-replace commitment. The Institution B case in Section 5 illustrates the model: a single transaction representing roughly 60% of the institution's daily cash transactions was modernized in seven days of focused development on top of an obsolete .NET 2.0 core, validating the pattern before broader migration [4].

Sub-vertical adaptations

Banks

For institutions on Fiserv, Jack Henry, or FIS cores, "modernizing the core" usually means modernizing around the core — building a parallel modern stack that gradually absorbs core functionality through APIs. Full core conversion remains a multi-year, eight-figure undertaking. Industry data shows mid-tier core conversions consistently run over budget and schedule.

Credit unions

Same pattern, with the additional consideration that core providers serving credit unions consolidate slowly. Conversion windows are limited.

Insurance carriers

Core insurance system replacements (PAS, claims, billing) are typically 36-to-60 month programs. Industry estimates put 2024 US insurance modernization spend at $132.86 billion, projected to reach $229.07 billion by 2029 [22]. Only roughly 10% of large insurance providers have modernized more than half of their core systems [23].

Fintechs

“Core” usually means the in-house ledger, payments engine, or risk engine. The question is when scale forces a rewrite. The honest answer for most fintechs is “once, between Series C and IPO.”

The matrix in one sentence.

If you can fund only one initiative this fiscal year, fund the channel layer; if you can fund two, add identity and fraud; if you can fund three, add the integration layer. Every other priority can wait one more cycle without becoming a board-level event. Channel and fraud cannot.

Section 3

The Executive Decision Framework

This section is for the CFO. The CTO already knows the technical case. What the CFO needs is a way to put a number on the cost of waiting another year, in language a board will accept.

The three decisions executives keep deferring

In our experience working with mid-size US financial institutions, three decisions show up on the executive agenda quarter after quarter and almost never get made. Each carries a measurable annual cost.

Decision 1: "Do we modernize the channel layer this year, or wait for next year's budget cycle?"

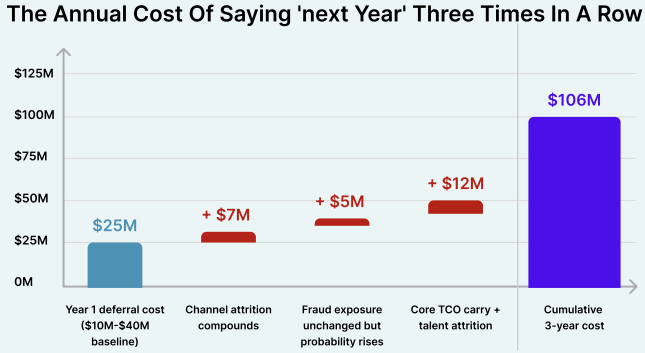
The deferral cost shows up in the deposit-attrition number, not the IT budget. For an institution losing 50 basis points of primary checking relationships per quarter to a digital-first competitor, with a $2,400 average LTV, every quarter of deferral costs roughly $1.2M per 100,000 customers. **Annual deferral cost: $4M–$15M** for a typical mid-size bank, escalating as the gap widens.

Decision 2: "Do we replace the rules-only fraud engine, or extend the existing vendor contract one more time?"

The deferral cost is asymmetric. In quiet years it is near zero. In a year with a major attack or a regulatory finding, it is multiples of the modernization cost. The IBM 2025 report puts US-domiciled financial-services breach exposure at $10.22 million on average [9]. **Expected annual deferral cost: $1M–$3M**, with significant variance.

Decision 3: "Do we begin the core modernization now, or wait until the current vendor contract is up for renewal?"

The deferral cost compounds slowly but reliably. Each year of delay adds approximately one year of carry on the legacy TCO multiplier (see F1), one additional cohort of retiring engineers, and additional integration debt. **Annual deferral cost: $5M–$25M**, with the upper end applying to institutions running undocumented or under-staffed cores.



For most institutions, the year-over-year cost of these three deferrals combined falls between $10M and $40M. That is the figure the CFO needs to walk into the board meeting with.

Building the modernization business case in board-readable language

A board that reads engineering specifications switches off in 90 seconds. A board that reads cash-flow analysis stays engaged.

The business case for modernization should be written as the latter, not the former.

The five line items every board case should include:

Run-the-bank reduction

Specific dollar reduction in maintenance and infrastructure spend, with the assumed timeline. Not “cost savings” — a number.

Delivery-velocity gains in revenue terms

Translate “ship features in three months instead of nine” into the revenue contribution of three additional product launches per year, multiplied by your average revenue contribution per launch. Show your work.

Risk-adjusted regulatory exposure

Probability-weighted cost of a consent-order remediation event over the next 36 months, with and without modernization. Boards understand expected value.

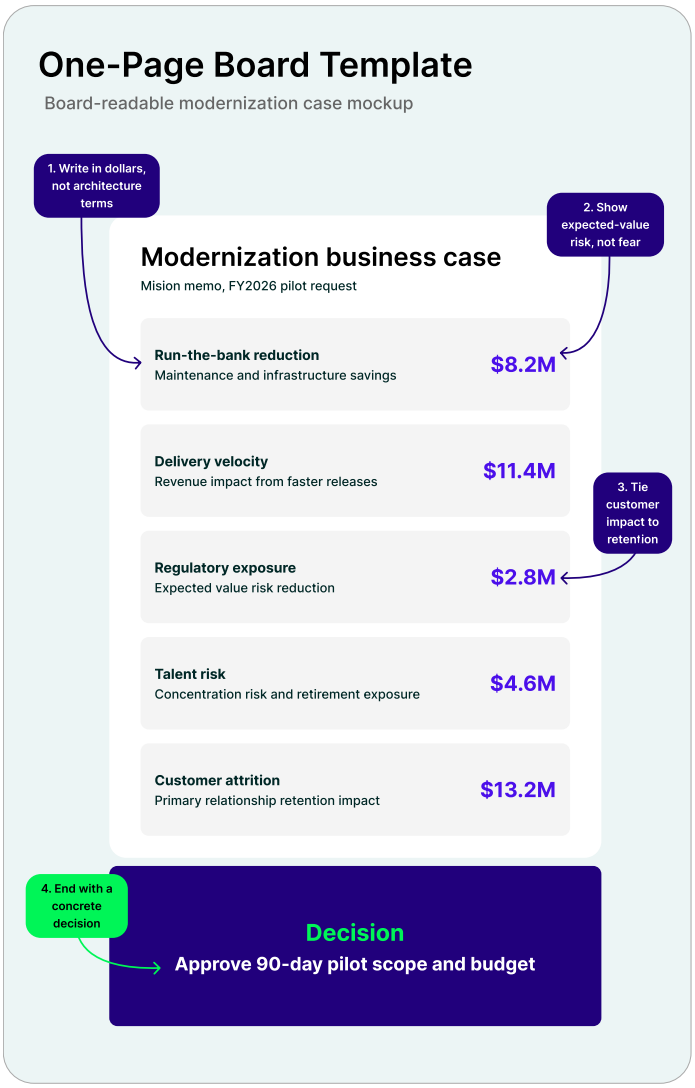
Talent and concentration risk

Specifically, the number of business-critical systems with two-or-fewer engineers who can maintain them, and the probability-weighted cost of a critical retirement event. (See Section 4.)

Customer-attrition reduction

Modeled deposit and primary-relationship retention impact at three scenarios: status quo, partial modernization, full channel modernization.

The case should be one page. Appendices can run to fifty pages; the board reads page one.



The false economy of “next year”

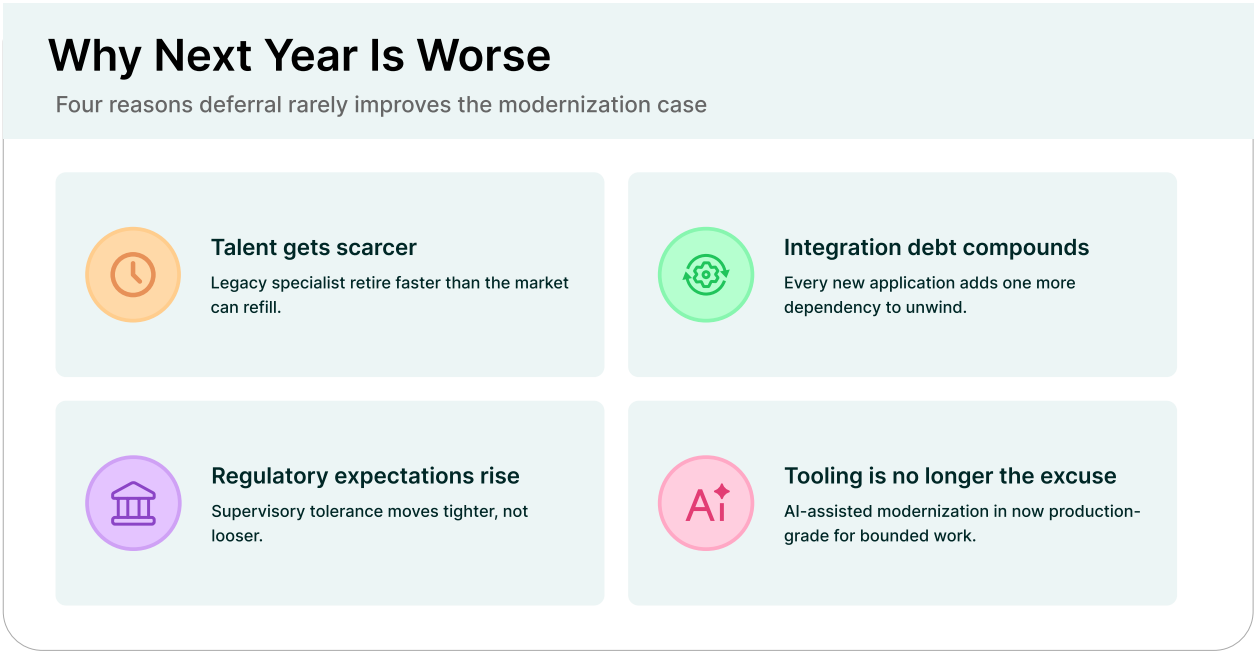
Every IT capital request that gets pushed to next year carries an implicit assumption: that next year’s environment will be more favorable for the work than this year’s. For modernization specifically, this assumption is almost always wrong, for four reasons.

Reason one: talent gets scarcer, not more abundant. The COBOL programmer cohort retires at roughly 10% per year [12]. The pipeline is not refilling. Each year of deferral is a year in which the work becomes harder to staff at any price.

Reason two: integration debt compounds. Each new system added to the existing landscape becomes another integration that must be maintained. The mid-size institutions we observe accumulate one-to-three new applications per year that further entangle with the legacy core. Deferring modernization means modernizing a more complex system.

Reason three: regulatory expectations escalate. The OCC’s posture on banks’ reliance on legacy systems has hardened steadily through the 2020s. The realistic forecast is that next year’s supervisory expectations will be tighter than this year’s, not looser.

Reason four: AI-augmented modernization is now in the picture. Three years ago, the rational deferral argument was “tooling will improve.” That argument is largely spent. AI-assisted refactoring, dependency mapping, and code-translation tools are now production-grade for the regulated, legacy-heavy environments banks and insurers operate in. Section 4 treats this in depth. The point here: the tooling is no longer the bottleneck. The decision is.



The cost-of-standing-still worksheet

Use this worksheet to estimate your institution's annual cost of standing still. The output is a single dollar figure suitable for the front page of a board case.

Estimated annual cost of standing still

Line item	Formula reference	Calculation	Your figure
A. Run-the-bank tax	See Formula 1	(Annual IT budget) × (% on maintenance) × 1.15 (mid-point tech-debt surcharge)	$_____
B. Hidden TCO uplift	See Formula 1	Reported core system cost × 2.4 (mid-point of Deloitte 3.4× multiplier minus reported portion)	$_____
C. Annual delivery drag	See Formula 2	(Releases delayed ≥6 months) × (Average release revenue contribution) × 0.5 (capture probability)	$_____
D. Regulatory exposure	See Formula 3	(P(consent order, 24m) × Estimated remediation cost) + (P(breach) × $6M financial-services average)	$_____
E. Concentration risk	See Formula 4	(Critical systems with ≤2 maintainers) × (Estimated cost of one critical retirement event)	$_____
F. Attrition-from-friction	See Formula 5	(Annual customer attrition) × (% attributable to digital experience) × LTV	$_____
Total		A + B + C + D + E + F	$_____

Honest caveats: lines D and E are probability-weighted; year-to-year variance will be high. Lines A, B, and F are recurring. Line C is recurring and growing. Treat the total as a directional figure suitable for board discussion, not a precision forecast.

The worksheet exists in calculator form online, with sensible defaults pre-filled by institution size and asset class.

Calculate your own cost of standing still — Devsu's ROI Calculator estimates your institution's annual modernization opportunity with VelX. It takes about ten minutes and produces a board-ready figure.

Section 4

The Talent Reality

The talent problem is the cost most institutions track least carefully and the cost that compounds most reliably. It is also the cost where intervention has the highest expected value, because the underlying labor market is moving against every institution that depends on legacy expertise.

Supply and demand: the data

The supply side of the legacy engineering market is in structural decline. The demand side is not. The result is a widening gap that is now visible in compensation data and recruiter postings.

Supply. Industry estimates put the average COBOL programmer's age at roughly 58, with approximately 10% of the cohort retiring each year [12]. Multiple secondary sources estimate that fewer than 2,000 COBOL programmers graduated worldwide in 2024, and that more than 85% of US universities have dropped COBOL from the curriculum since the 1990s [13]. RPG (the AS/400 / IBM i programming language) is in similar or worse condition; the typical RPG programmer is now around 70 years old, with industry estimates projecting near-total retirement of the RPG cohort by 2030 [13].

Demand. COBOL still underpins approximately 43% of US banking systems and 95% of ATM transactions [11]. None of these systems are being decommissioned on a timeline that matches the retirement curve. Industry estimates project sustained demand for legacy maintenance and modernization roles through at least 2035 [13].

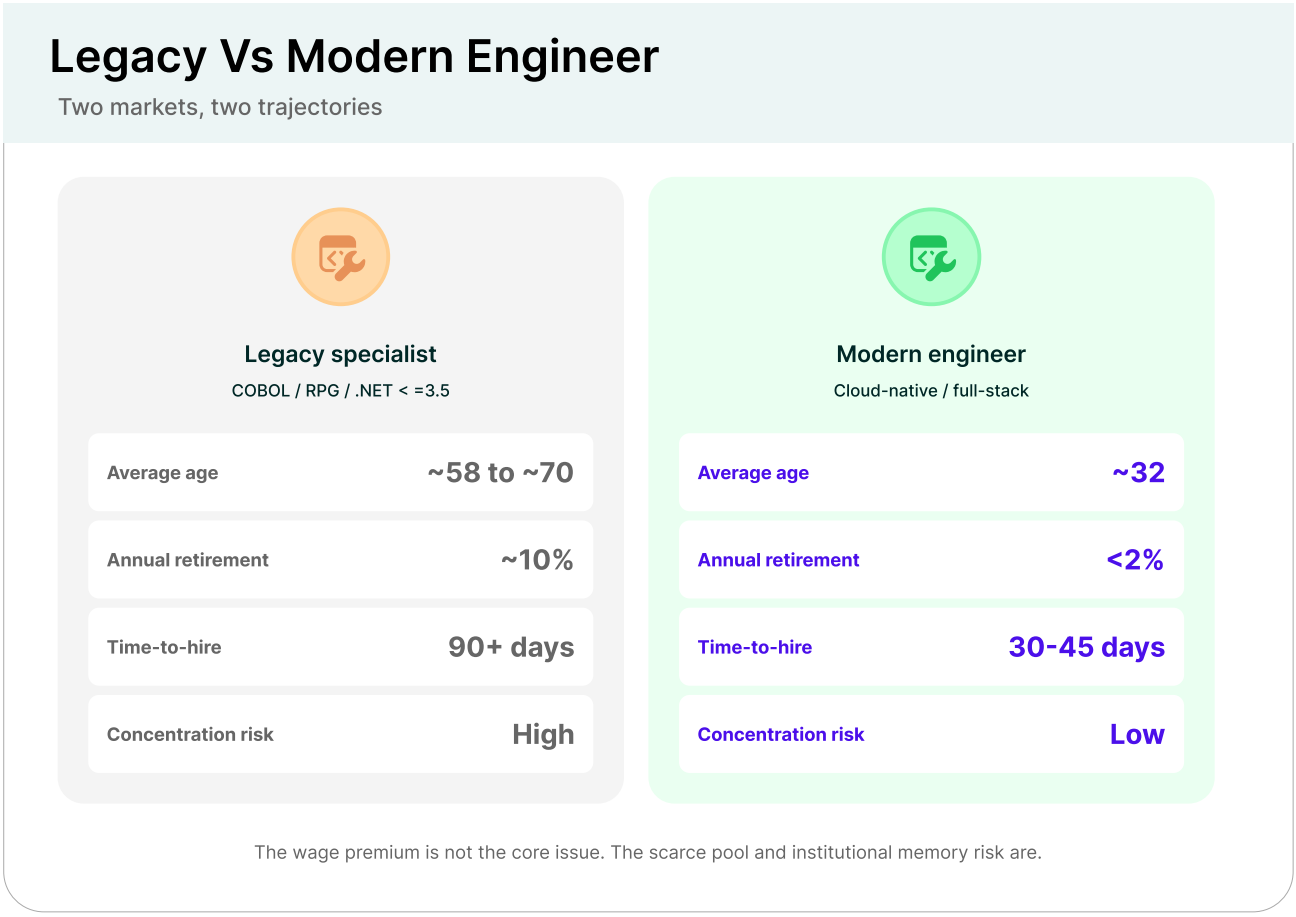
Compensation signal. Salary data from public job-board aggregators in 2025 show COBOL maintenance roles in the $86K–$121K range, with modernization-consulting roles reaching $150K, and significant geographic variation. The Bureau of Labor Statistics' Q1 2026 Employment Cost Index shows continued positive wage growth in finance and insurance occupations, consistent with the broader pattern of professional/management compensation outpacing CPI [24].

For comparison, modern-stack engineers (cloud-native, full-stack, ML/MLOps) command similar or higher compensation but are recruited from a vastly larger pool. The wage premium for legacy specialists is not dramatic; what is dramatic is the difficulty of finding any legacy specialist at any price.

Legacy vs. modern engineering talent market (2025-2026)

Dimension	Legacy (COBOL / RPG / .NET ≤3.5)	Modern (Cloud-native / Full-stack)
Average age (US)	~58 (COBOL); ~70 (RPG)	~32
Annual retirement rate	~10%	<2%
New entrants per year (US)	<2,000 globally (COBOL) [VERIFY]	Tens of thousands
Typical compensation range (USD)	$86K-$150K	$110K-$200K+
Time-to-hire (days, median)	90+	30-45
Concentration risk	High; often single-individual dependencies	Low

Sources: Fujitsu / IBM (2020) on COBOL demographics; aggregated recruiter postings 2025; BLS Employment Cost Index Q1 2026. See Appendix B, refs [12], [13], [24].



Wage inflation and the budget consequences

Two patterns are worth noting for budget planning.

First, wage inflation in financial-services engineering has moderated from its 2021–2023 peak but remains positive, with senior-engineer base growth running roughly 6–10% year-over-year across most specializations as of Q1 2026 [25]. For specialized roles — DevOps/SRE/platform, security engineering — the rate is higher, 10–16% year-over-year.

Second, scarcity premiums for legacy specialists are unstable. They spike when an institution faces a forced modernization (e.g., consent-order deadline) and the available specialist pool concentrates around the highest-paying engagement. Budgeting against last year's rate is a poor planning assumption; budget against a rate ~20% higher and a time-to-hire significantly longer than for modern roles.

For a mid-size institution with 30–50 engineers on legacy maintenance, the realistic three-year talent-cost trajectory is $2M–$8M of incremental wage and contractor expense above baseline, with the variance driven primarily by retirement events and forced-modernization episodes.

Why waiting makes the talent problem worse

Three mechanisms compound:

Retirements continue regardless of institutional intent. Each year, ~10% of the available legacy specialist cohort exits the workforce. That share is not recoverable; the labor market does not back-fill at meaningful scale.

Institutional knowledge degrades. Legacy systems with two-or-fewer maintainers are one retirement event away from undocumented black-box status. We see this pattern weekly. By the time the institution decides to modernize, the engineers who remember the system's quirks are gone.

Modernization itself requires legacy specialists. The first phase of any serious modernization — discovery and dependency mapping — requires people who can read the legacy code. Waiting until the talent has retired means starting modernization without the people who could have made it succeed.

The implication is uncomfortable.

The optimal time to modernize is when you still have the legacy specialists to inform it, not when you no longer do.

Four credible responses (with honest tradeoffs)

There are four serious approaches to the talent problem. Each has real strengths and real failure modes. We represent them as fairly as we can; Devsu has commercial interest in only one of them, but we are not in the business of selling against institutions whose situation calls for one of the other three.

Approach 1: Offshore traditional outsourcing

Strengths. Known model, established providers, predictable cost structure, large pool of intermediate-skill engineers. Works particularly well for sustained maintenance of stable legacy systems.

Failure modes. The people who actually understand the legacy stack are typically not the ones offshore providers can scale; quality variance is high; institutional knowledge transfer is difficult; effective velocity rarely matches what the cost-per-hour ratio implies.

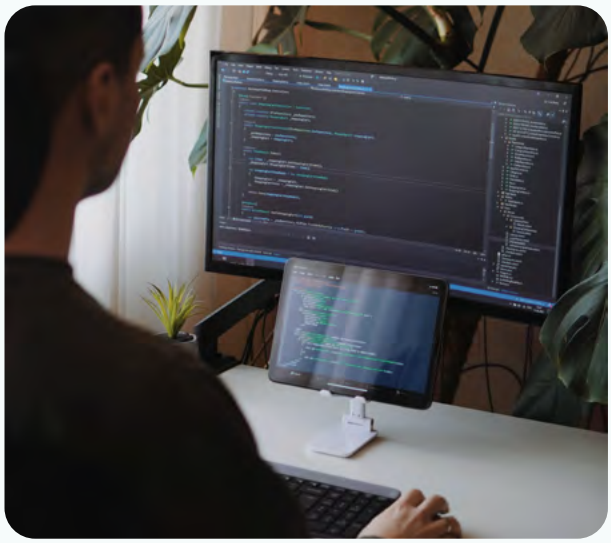
When to choose. Stable maintenance workload, where speed of change is not the constraint and predictable cost is the goal.

Approach 2: In-house build and retain

Strengths. Maximum institutional knowledge retention, full control of tooling and standards, alignment with risk and security culture, no third-party dependency.

Failure modes. Recruiting time-to-fill is long for legacy roles; the institution becomes a training ground that loses senior engineers to better-paying offers; achieving modern delivery velocity requires investment in tooling that most mid-size institutions cannot justify on a single program.

When to choose. Institutions with genuine engineering cultures and the willingness to invest 18–24 months in building one, before expecting velocity returns.

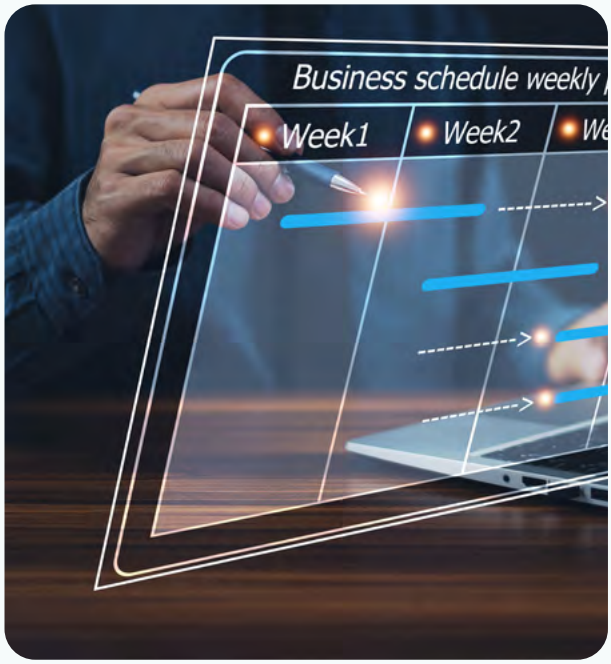


Approach 3: Packaged platform replacement

Strengths. Vendor takes on the modernization risk; clear contract structure; established product roadmap; configuration rather than custom development.

Failure modes. Configuration limits often surface only mid-implementation; integration with non-platform systems remains expensive; vendor lock-in is severe; the implementation is the modernization (12–36 months of program risk concentrated in a single vendor relationship).

When to choose. Institutions whose business processes genuinely match the packaged platform and who can tolerate 24–36 months of program execution risk. Increasingly common for credit unions selecting modern core providers and for insurance carriers selecting modern PAS.



Approach 4: AI-augmented engineering teams

This is the approach Devsu's VelX framework implements, and it is fair to disclose that we have a commercial interest. We will represent it on its merits and its limits.

Strengths. AI-assisted dependency mapping, code translation, and refactoring meaningfully reduce the time required for legacy specialists to do discovery and modernization work; modern engineers become productive on legacy stacks faster than they would with documentation alone; the human-in-the-loop pattern preserves engineering judgment for the work that genuinely requires it. In the Institution B case, AI-assisted discovery of an undocumented .NET 2.0 codebase reduced an analysis cycle that would normally take three-to-six months to under three weeks [4].

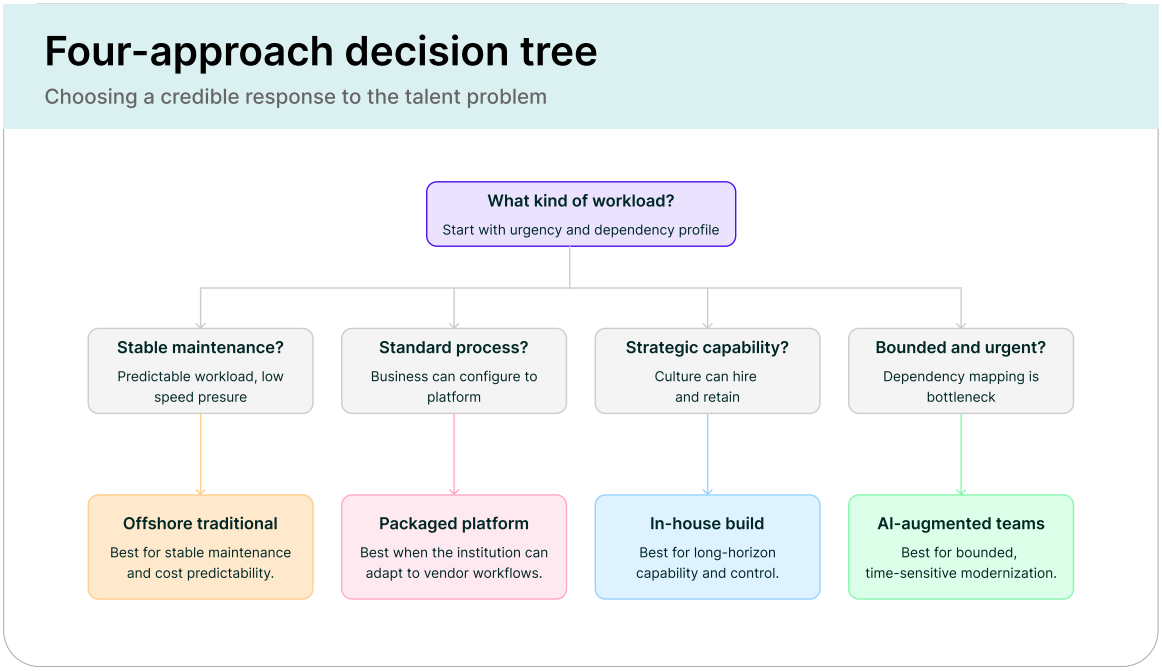
Failure modes. The approach is only as good as the human engineers in the loop; AI-generated refactoring still requires senior review for security, regulatory compliance, and edge-case correctness; the tooling does not replace institutional knowledge for genuinely complex business logic; and like any new tooling, it carries adoption risk and an integration cost into existing engineering processes.

When to choose. Institutions with bounded, time-sensitive modernization work where the bottleneck is dependency-mapping and code-comprehension speed, and where the engineering culture is willing to adopt AI-assisted patterns. Less appropriate for deeply customized, business-logic-heavy environments where the underlying logic is not documented and the surviving experts are unavailable.

Comparison of four talent-strategy responses

Approach	Best for	Realistic time to first value	Concentration risk	Per-engineer effective velocity
Offshore traditional	Stable maintenance, predictable workload	3-6 months ramp	Medium	0.6× to 0.9× of in-house
In-house build & retain	Long-horizon strategic capabilities	18-24 months	Low (over time)	1.0× baseline
Packaged platform	Standard processes, willing to configure	18–36 months	Vendor-concentrated	N/A — consumption model
AI-augmented engineering	Time-sensitive bounded modernization	4–12 weeks	Low	1.4×–2.0× on suited workloads

Sources: Devsu engagement data; comparison structure adapted from McKinsey, Forrester, and Gartner analyses of modernization sourcing models (2024–2025). Velocity ranges are directional and depend on workload type.



The honest summary

There is no single right answer to the talent question. There is a wrong answer: assume next year's labor market will be more favorable than this year's. For legacy financial-services engineering, it will not be.

Section 5

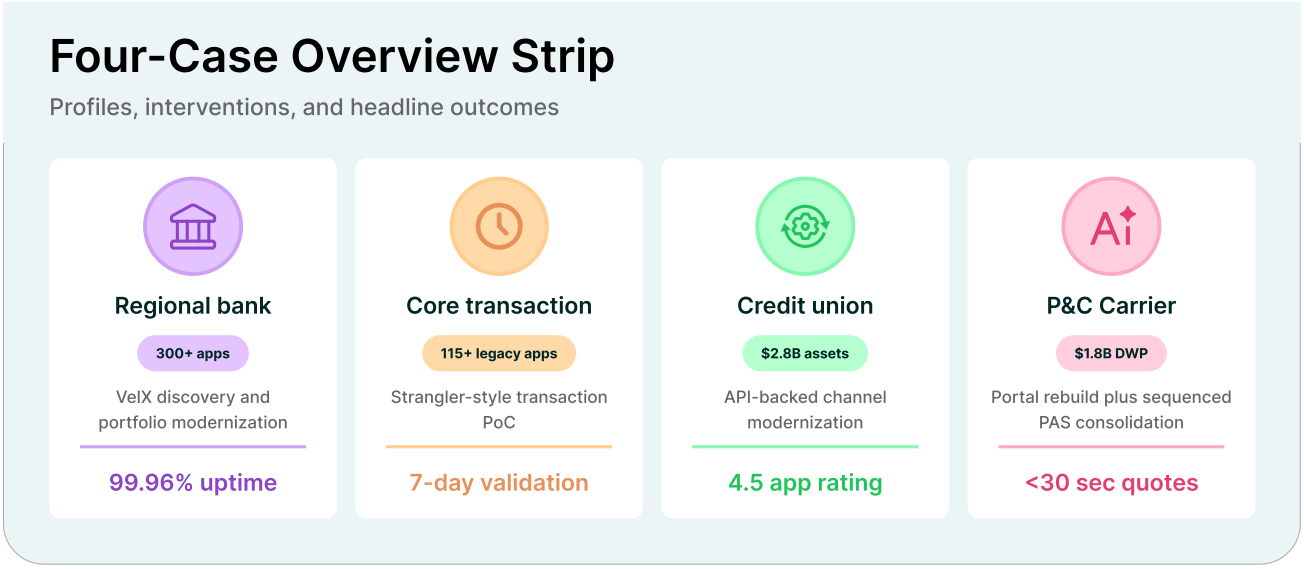
The Real Stories

This section presents one mini-case per vertical: a regional bank, a credit union, an insurance carrier, and a Series B+ fintech. We are honest about sourcing. The two bank stories are drawn from Devsu's engagement records — institutional names have been withheld; these are operating institutions in the Americas with public regulators.

The credit union and insurer stories are composite scenarios assembled from publicly-reported industry patterns and analyst observations, clearly marked as such. The point of these stories is not to claim universal applicability. It is to give the reader a concrete reference for what modernization actually looks like inside a real institution.

Four-Case Overview Strip

Profiles, interventions, and headline outcomes



Institution A — Top-five Regional Bank

(Devsu engagement, 2024–2025)

Context. A century-old top-five regional bank in the Americas, operating with more than 300 internal applications accumulated through organic growth and M&A. Many were undocumented, unsupported, or duplicated function. Internal teams estimated $80M in annual losses tied to inefficient systems, duplicated work, unnecessary maintenance, and unplanned downtime. Uptime had drifted to 98.9%, well below the 99.9% industry benchmark, and regulatory audits were chronically painful due to documentation gaps.

What the bank did. The bank engaged Devsu to deploy VelX, an AI-augmented modernization framework, across the application portfolio. The first phase — discovery and dependency mapping — was completed in under three weeks for assessments that typically take three to six months on this scale of estate. The discovery phase identified that approximately 35% of applications were redundant or outdated, with no centralized visibility that would have allowed the bank to confidently decommission them.

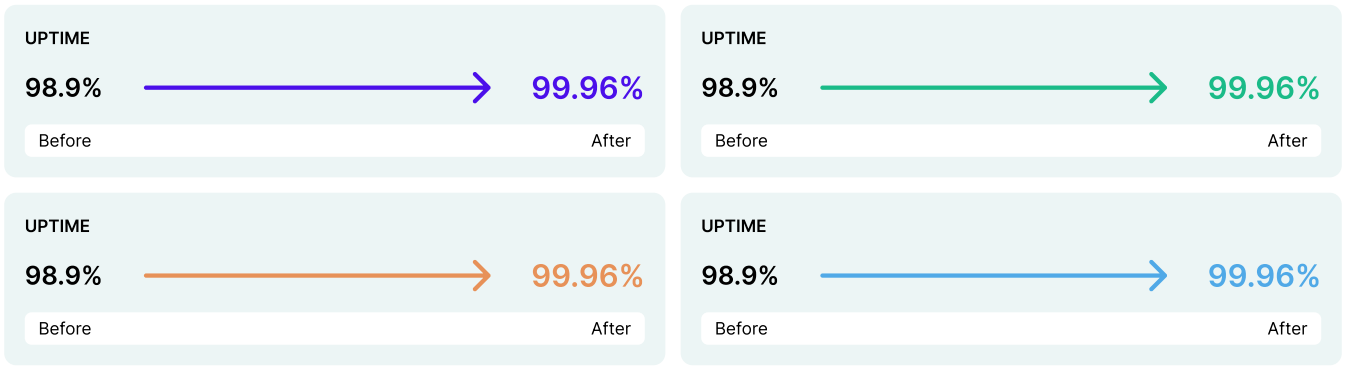
The work proceeded in three phases: AI-assisted discovery (90% reduction in analysis time), architecture and planning (60% faster, planning cycles compressed from 10 weeks to 4), and implementation (40% reduction in delivery timelines, with 8-month cycles compressed to 12–14 weeks).

Outcomes. Estimated annual savings of $24M–$30M, driven by system decommissioning, reduced rework, and fewer outages. Uptime recovered to 99.96%. Critically, the bank now has a structured technical-debt report covering the full estate — the artifact that future modernization decisions will be made against.

What this case is and is not. It is a real engagement, in production, at a top-five regional institution operating in regulated environments. It is not directly transferable to every US institution. The bank in question runs a portion of its core processing on COBOL applications and a portion on .NET-based middleware; institutions with different stack composition will see different velocity ratios.

Institution A — Before / After Dashboard

Anonymized engagement, Q4 2024 → Q3 2025



Portfolio discovery created the technical-debt report used for future modernization decisions.

Institution B — Core Transaction Modernization

(Devsu engagement, 2025)

Context. A Latin-American institution with more than 100 years of operating history and an IT estate including 115+ legacy applications. The most acute pain point was a single application running on the obsolete .NET 2.0 framework, reachable only through pre-2008 versions of Internet Explorer, hosted on pre-2008 servers, and unpatched for security for approximately 8–10 years. The application processed approximately 60% of the bank's daily cash transactions. One internal engineer retained working knowledge of the codebase, and that knowledge was incomplete.

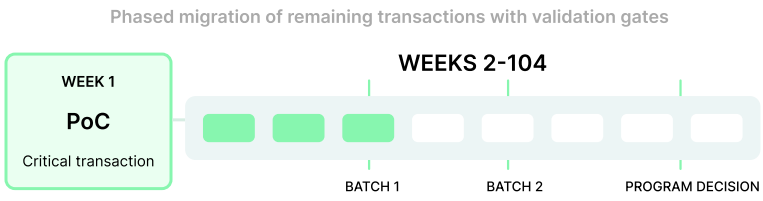
What the bank did. Rather than pursuing a full rewrite — which the bank's analysis projected would cost more than the bank could justify — Devsu and the bank executed a proof-of-concept modernization on the most critical transaction (deposit-to-current-account) using the VelX framework with human-in-the-loop AI assistance.

Outcomes. The single critical transaction was modernized in seven days of focused development time. The application was migrated from .NET 2.0 to .NET 3.5 (a controlled, stability-first upgrade) and the modernized transaction ran in modern browsers on modern hardware for the first time in more than a decade. Bank tellers performed the transaction live, with real-time data reflection across the bank's other legacy systems, validating both the migration and its integration with the surrounding estate. The bank now has a structured technical debt report covering the remaining ~114 transactions, against which a phased migration is being planned.

What this case is and is not. It is a single proof-of-concept that demonstrated a pattern. The pattern — strangler-style modernization of one critical transaction at a time, AI-assisted discovery and refactoring, human-in-the-loop engineering — is the model. The seven-day cycle is not. Production-level modernization of all 100+ transactions is a multi-year program, and the bank has not committed to any single timeline.

Institution B — Strangler Pattern Timeline

Pattern proven before broader scope is committed



Mid-size credit union (composite scenario)

Context. A $2.8B-asset credit union serving roughly 180,000 members across three states, with a J.D. Power-recognized staff and a member NPS that consistently outpaces nearby community banks. The credit union runs a packaged core platform from one of the established credit-union core providers. The pain point is not the core. It is the channel and the integration layer: the mobile app is rated 3.4/5.0 in app stores, member-facing online banking lags megabank competitors by roughly two years on feature parity (no real-time external transfers, no integrated subscription management, no integrated savings goals), and the average new-member acquisition cost has climbed roughly 35% over three years.

What the institution does. The credit union pursues a 9-month, $4.5M channel modernization program: a new mobile and web shell built against an API gateway sitting in front of the existing core, with subscription management, savings goals, and improved external-transfer capability shipped in three quarterly releases. The core itself is not touched.

Outcomes (modeled). App rating recovers to 4.5/5.0 within 6 months. New-member acquisition stabilizes. Estimated three-year retention impact: 80–150 basis points of improved primary-relationship retention, translating to $7M–$13M of preserved deposits and fee income at the institution's current LTV.

What this scenario is and is not. It is composite, drawn from the Cornerstone Advisors / Alkami 2025 Digital Banking Performance Metrics Report benchmarks [15] and from industry retention research. It is presented as a credible mid-case for a $2–5B credit union in 2026. Actual results vary substantially with execution quality, vendor selection, and member demographics.

Mid-market P&C insurance carrier (composite scenario)

Context. A US-domiciled mid-market P&C insurance carrier with $1.8B in direct written premium across personal auto, homeowners, and small commercial. The carrier operates four policy administration systems acquired through a series of acquisitions over 15 years; two of the systems run on platforms whose original vendors no longer offer active support. The agent-facing portal — now the channel through which 70% of new business arrives — requires 90 seconds to return a homeowners quote; competitors return quotes in under 30 seconds.

What the institution does. Rather than pursuing a full PAS consolidation (which industry experience indicates would be a 36–60 month program at significant capital cost [22], the carrier pursues a sequenced approach: (1) replace the agent portal and quoting layer with a modern frontend that pulls rates from each underlying PAS via an integration layer (12 months); (2) consolidate two of the four PAS instances onto a single modern platform on a 30-month track; (3) defer further consolidation pending performance results.

Outcomes (modeled). Quote-return time falls below 30 seconds within 12 months, with measurable agent-distribution share recovery in years two and three. The 30-month PAS consolidation runs over original schedule by ~25% (consistent with industry experience for projects of this scale) and modestly over budget. The deferred-consolidation decision is the single best decision the carrier makes — the 30% of cases that turn out to require the deferred system are exactly the cases that would have caused the worst pain in a forced full consolidation.

What this scenario is and is not. It is composite, based on Celent's 2024 P&C Insurance IT Pressures & Priorities research [21] and the industry-wide observation that only roughly 10% of large insurers have modernized more than half their core systems [23]. The numbers should be treated as directional.

Institution C — Series B+ consumer-credit fintech
(composite, anchored to real engagement)

Context. A US consumer-credit fintech, four years post-Series B, with approximately 600,000 active borrowers and $1.4B in originated loans. The product runs on a custom-built application stack that scaled well to ~200,000 users but is showing wear at the current scale: average loan-application latency has crept from under 2 seconds to over 8; engineering velocity on new features has fallen by half; and a recent BSA/AML examination identified three matters requiring attention related to transaction monitoring.

What the institution does. The fintech does not rewrite the core. It pursues a focused 90-day modernization of the loan-application workflow specifically, using AI-assisted refactoring of the existing codebase. A workflow that previously required 144 engineering hours per release cycle is reduced to roughly 40 hours, freeing the engineering team for the AML remediation and for new product work [4].

Outcomes (modeled). Loan-application latency returns to sub-2-second medians. Release-cycle engineering effort is reduced by approximately 70%. AML remediation completes within the supervisory timeline. NPS rises ~30% on the back of faster decisions and fewer errors.

What this scenario is and is not. The 144-to-40-hour and ~30% NPS lift figures come from a real Devsu engagement with an anonymized consumer-credit institution [4]. The framing — fintech rather than bank — is illustrative. The general pattern (workflow modernization, AI-assisted refactoring, ~3-month cycle) is increasingly common across institutions in the early-stage growth phase.

What the four stories have in common

None of these institutions began with a multi-year program. Each began with a bounded pilot scoped to 90–180 days of work on a specifically-identified pain point. Each treated the pilot as a hypothesis test about whether the AI-augmented modernization model produced the velocity it claimed. In each case, the pilot's outcome — not the vendor's promise — was what determined the size of the follow-on program.

That sequence is the subject of Section 6.

Section 6

The Path Forward

The argument of this ebook is that the cost of waiting is now measurable and, in most cases, exceeds the cost of starting.

This section addresses the obvious follow-up: if we are going to start, what does "starting" actually look like, without committing to a multi-year program?

The bounded pilot: what to pick

The single most important decision in any modernization program is the choice of the first pilot. Pick well, and the rest of the program proceeds on the strength of demonstrated outcomes.

Pick badly, and the program either fails outright or — more commonly — succeeds on the wrong dimension and produces no momentum for the broader effort.

Five characteristics of a well-chosen pilot:

Genuinely critical, not symbolic.

A pilot that touches a system the executive team does not care about will produce a successful pilot and no follow-on. Pick a system whose successful modernization changes a P&L line item or a regulatory exposure.

Bounded in scope.

A pilot that requires modernizing seven systems to demonstrate value is not a pilot. Pick a workload whose business value is measurable in isolation. The Institution B case is a good reference: one transaction, representing 60% of daily cash transactions, modernized fully — discovery through production validation — before broader scope was discussed.

Honest about dependencies.

A pilot that pretends it has no upstream or downstream dependencies will discover them in production. Map the dependencies before scoping the pilot, not during.

Time-boxed at 90–180 days.

Beyond 180 days, organizational attention drifts and the pilot becomes a program. Inside 90 days, the work is too small to demonstrate the velocity differential the vendor claimed.

Has a "did it work?" criterion that the CFO accepts.

Not "we shipped on time." Specifically: "the metric we said would move, moved by the amount we said." Common metrics: feature delivery time on the modernized component, breach probability score, application latency, agent quote-return time, NPS.

Pilot scorecard

	Criterio	Score	Example
1	**Critical, not symbolic** Changes a P&L line item or regulatory exposure	R Y G	Score before funding
2	**Bounded in scope** Value measurable without seven-system dependency	R Y G	Score before funding
3	**Honest dependencies** Upstream and downstream systems mapped first	R Y G	Score before funding
4	**Time-boxed** 90-180 days, not a disguised program	R Y G	Score before funding
5	**CFO-accepted metric** The agreed metric moves by the agreed amount	R Y G	Score before funding

The 90-day proof-of-value approach

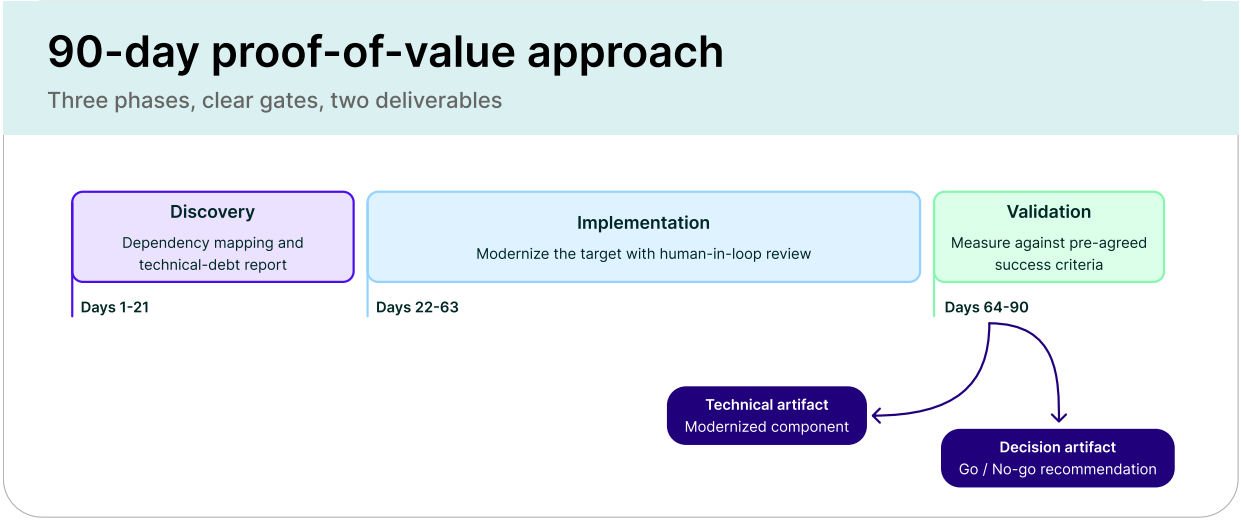
A 90-day proof of value, executed properly, answers four questions:

1. **Is the underlying tooling — AI-assisted discovery, refactoring, dependency mapping — actually production-grade for our specific stack?** Vendor demos are not evidence. Your codebase is.
2. **Can the chosen vendor's engineers work effectively with our institutional context — our security posture, our regulatory environment, our internal review processes?** This is a cultural and process question more than a technical one.
3. **Will the modernized component perform under our actual load, against our actual data, with our actual dependencies?** Not in a sandbox. In a production-equivalent environment.
4. **Does the cost-velocity ratio of the broader program look defensible against the worksheet from Section 3.4?** If the 90-day pilot does not extrapolate to a clear board case, the broader program does not get funded — and that is the right outcome.

A typical 90-day proof of value runs in three phases:

- **Days 1–21: Discovery and dependency mapping.** AI-assisted analysis of the legacy code, structured technical-debt report, identification of the highest-value modernization target within the broader scope.
- **Days 22–63: Implementation.** Modernization of the specific target — a transaction, a workflow, a service — with human-in-the-loop engineering review at each stage.
- **Days 64–90: Production validation and decision.** Live execution against production data, structured measurement against pre-agreed success criteria, and a written go/no-go recommendation for the broader program.

The deliverable at day 90 is two documents: a technical artifact (the modernized component, in production or production-equivalent) and a decision artifact (a recommendation for or against the broader program, with sourced numbers).



The questions executives should ask any vendor (including Devsu)

Modernization vendors all sound the same in the first meeting. The differences emerge in answers to specific questions. We list these without hedging — they apply to Devsu the same as to any other vendor, and an executive should ask them all.

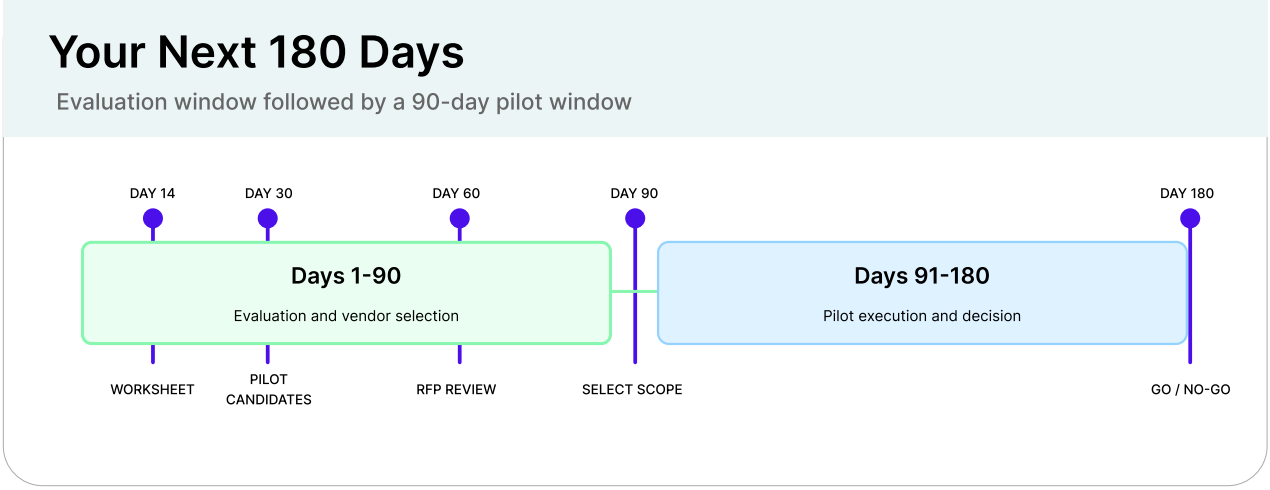
1. **"Show me a case where this approach failed. What did you learn, and what would you do differently?"** Vendors who cannot answer this question have either never failed or cannot describe a failure honestly. Both are concerning.
2. **"What is the role of human engineers in your AI-augmented model? Where specifically do they make decisions the AI cannot?"** Vendors whose answer is "the AI does it" are over-claiming. Vendors whose answer is "humans do most of it" have not built differentiated tooling.
3. **"How do you handle regulatory review and audit traceability for AI-generated code?"** This is now an OCC and NCUA examiner question. The answer should reference specific controls, code-review trails, model governance, change management — not generalities.
4. **"What is the smallest engagement you will accept, and the shortest?"** Vendors whose minimum engagement is 12 months and seven figures are selling programs, not pilots. For a first engagement, a vendor unwilling to do bounded 90-day work is the wrong vendor.
5. **"How will you transfer institutional knowledge to our internal team during and after the engagement?"** Vendors whose answer is "documentation at the end" produce dependency, not capability. The right answer involves continuous pairing and structured handoff during the engagement.
6. **"What does your engagement look like if your AI tooling underperforms on our codebase?"** Honest vendors have a backup plan. Vendors who insist their tooling will work universally are over-promising.
7. **"Who will actually do the work? What is the seniority mix, and how is it priced?"** Modernization work in regulated environments is not junior-engineer work. The seniority mix is the single best predictor of engagement quality.

What the next 90 days should look like

For institutions persuaded that the cost of standing still is real but unsure where to start, a reasonable next-90-days plan looks like this:

- **Days 1–14:** Run the worksheet from Section 3.4 against your institution's actual data. Generate a defensible total-cost-of-standing-still figure. Walk it through the CFO and CTO independently.
- **Days 15–30:** Identify the three highest-impact bounded modernization candidates against the priority matrix in Section 2. Score them against the five pilot characteristics in 6.1.
- **Days 31–60:** Issue a structured RFP to two-to-three vendors against the questions in 6.3. Insist on bounded 90-day pilot scoping. Reject vendors who cannot scope to that constraint.
- **Days 61–90:** Select a pilot vendor and pilot scope; align internal stakeholders (engineering, risk, compliance, the CFO); secure the pilot budget. Begin the pilot at day 91.

This plan does not commit the institution to a multi-year program. It commits the institution to having an actual answer, by day 180, to the question: "is the cost of standing still bigger than the cost of moving?"



Closing

This document has argued that standing still has a cost, that the cost is measurable, and that the cost has now risen to the point where deferral is harder to defend than action. We do not pretend that modernization is risk-free. It is not. Programs run over schedule and over budget. Tooling underperforms. Vendors over-promise. Internal cultures resist. Failures happen.

What has changed in 2026 is the asymmetry. Three years ago, the case for waiting another cycle was credible — tooling was immature, talent was less scarce, and the regulatory posture was softer. None of those conditions still hold. The honest comparison is no longer "do nothing vs. take a risk." It is "take the risk of action vs. take the risk of inaction." Each institution will weigh those risks differently. Most of them will conclude, after running the math, that inaction is the larger risk.

devsu

Book your 30-minute Modernization Strategy Session with Devsu's engineering leadership.

This is a conversation with engineers, not sales. The agenda is your actual codebase, your actual constraints, and an honest read on whether a bounded pilot makes sense for your institution.

Visit devsu.com

Appendix A

Formula Index

All formulas used throughout this ebook in one place. Each formula is anchored to its first appearance in the text and to the cost-of-standing-still worksheet in Section 3.4.

Run-the-bank tax (with hidden TCO)

Formula 1

(Annual IT budget × % allocated to maintenance) × (1 + tech-debt surcharge)

Hidden TCO multiplier = Reported core system cost × ~3.4

In plain English: How much of your IT budget is being consumed just to keep the lights on, once you account for the tech-debt drag on every new project plus the costs hidden in vendor management, integration, and compliance overhead.

Annual delayed-delivery cost

Formula 2

(Releases delayed by ≥6 months) **× (Avg. annualized revenue contribution per release)** **× (Probability competitor ships first)**

In plain English: The revenue your institution forgoes each year because legacy architecture ships features in 9–18 months while modern competitors ship in 3–4.

Annual regulatory exposure

Formula 3

P(consent order, 24m) × Estimated remediation cost **+ P(breach) × Average breach cost** **+ P(regulator-driven growth restriction) × Foregone revenue**

In plain English: Expected-value cost of compliance failure: even small probabilities applied to large remediation events produce significant numbers.

Talent concentration risk & carry cost

Formula 4

Concentration risk =

(Critical systems with ≤2 internal maintainers)

÷ (Total business-critical systems)

Annual talent carry cost =

(Wage premium for legacy specialists vs. modern engineers) × Headcount

+ (Estimated cost of one critical retirement event in next 24 months)

In plain English: How dependent your institution is on a handful of irreplaceable people, and the financial exposure that creates.

Annual attrition-from-friction cost

Formula 5

(Annual customer attrition × % attributable to digital experience × LTV) **+** **(Net new customers lost to digital-first competitor × LTV)**

In plain English: The deposit and primary-relationship value bleeding off through a sub-par digital channel — often booked under "marketing" rather than "IT"

Total annual cost of standing still

Formula 6

Total = F1 (run-the-bank tax)

+ (Hidden TCO uplift)

+ F2 (delivery drag)

+ F3 (regulatory exposure)

+ F4 (concentration risk component)

+ F5 (attrition-from-friction)

In plain English: The single dollar figure suitable for the front page of a board case. See the worksheet in Section 3.4.

Appendix B

References

All data cited in the body of this ebook is sourced below. Citations use numbered superscript [n] in the text and resolve to the corresponding entry here. Formatting follows APA-7 conventions for industry publications and corporate research reports.

Where a primary source is still being verified, the entry is marked [Verification in progress]. Where data is internal to Devsu, the engagement reference is anonymized at the institution's request.

References cited

[1] McKinsey & Company. (2024). Banking technology spend and the cost of technical debt. McKinsey Global Banking Practice. Retrieved from McKinsey research portal.

[2] Deloitte. (2024). Banking technology survey: True total cost of ownership of legacy core systems. Deloitte Center for Financial Services. [Verification in progress: locate primary Deloitte publication rather than secondary industry summary.]

[3] McKinsey & Company. (2025). Global Banking Annual Review 2025. McKinsey Global Banking Practice.

[4] Devsu. (2025). Anonymized client engagement records: VelX modernization framework deployments at Institutions A, B, and C (financial-services sector, Americas), 2024–2025. Devsu internal engagement records.

[5] Goodwin Procter LLP. (2024, August). Federal banking regulator enforcement actions, June 2023 – June 2024: analysis of OCC, Federal Reserve, and FDIC public filings. Goodwin Procter financial-services regulatory practice.

[6] Office of the Comptroller of the Currency. (2024, January). Consent order: Blue Ridge Bank, N.A. (Information Technology Control Program). OCC Enforcement Action Database.

[7] Office of the Comptroller of the Currency. (2024, February). Cease and desist order with civil money penalty and GLBA agreement: City National Bank, Los Angeles. OCC Enforcement Action Database.

[8] IBM Security & Ponemon Institute. (2024). Cost of a data breach report 2024. IBM Corporation.

[9] IBM Security & Ponemon Institute. (2025). Cost of a data breach report 2025. IBM Corporation.

[10] Reuters. (2017). Banks scramble to fix old systems as IT "cowboys" ride into the sunset. Reuters Business News. Cited in subsequent industry analyses; 2022 industry estimates range 220–800 billion lines globally.

[11] COBOL Cowboys et al. (2024). The COBOL footprint in US financial infrastructure: industry estimates. Industry analyses citing Reuters and Fujitsu primary sources.

[12] IBM & Fujitsu. (2020). COBOL programmer demographics: average age, retirement rate, and US retailer reference case. Fujitsu Global blog and IBM mainframe analyst commentary.

[13] Aggregated recruiter postings, industry compensation surveys, and modernization labor-market analyses, 2024–2025. Includes public job-board data on COBOL maintenance and modernization consulting roles.

[14] Fujitsu Global. (2020). Frequently-cited case study: US retailer with 2,500 stores, ~20 million lines of COBOL, two maintainers. Fujitsu modernization blog.

[15] Cornerstone Advisors & Alkami. (2025, April). Digital banking performance metrics report (6th annual edition). Cornerstone Advisors research, sample average $4.7B in assets.

[16] Industry retention research, aggregated. Estimates of customer acquisition cost vs. retention cost in US retail banking; Reichheld retention-economics framework (5% retention improvement → 25%–95% profit lift).

[17] Office of the Comptroller of the Currency. (2024, October). Consent order with civil money penalty: TD Bank, N.A. — BSA/AML transaction monitoring. OCC Enforcement Action Database.

[18] Office of the Comptroller of the Currency. (2024, March). $250 million civil money penalty: JPMorgan Chase Bank, N.A. — trade surveillance and oversight gaps. OCC Enforcement Action Database; SEC 8-K filing.

[19] National Credit Union Administration. (2025). NCUA 2025 supervisory priorities letter. NCUA Office of Examination and Insurance.

[20] Cornerstone Advisors. (2025). What's going on in banking 2025: annual industry leader survey. Cornerstone Advisors research.

[21] Celent. (2024). Dimensions: P&C insurance IT pressures & priorities 2024 — North American edition. Celent insurance research.

[22] Industry market sizing. (2024–2025). US insurance modernization spend forecasts, 2024–2029. Aggregated from public analyst reports.

[23] Novarica (now part of Aite-Novarica Group). (2024). Insurer core-system modernization survey: share of large carriers having modernized more than half their core systems. Industry survey cited across analyst publications.

[24] US Bureau of Labor Statistics. (2026, Q1). Employment Cost Index — finance and insurance occupations. BLS Office of Compensation and Working Conditions.

[25] Engineering compensation benchmarks, 2025–2026. Aggregated from public salary databases, recruiter postings, and industry compensation trackers. Specific provider citations available on request.

About Devsu

Devsu is a global technology partner helping organizations accelerate product development, modernize infrastructure, and build future-ready digital experiences. With teams across the Americas and clients worldwide, we specialize in delivering solutions that combine speed, quality, and intelligence.

We believe in sharing knowledge with the community, which is why we created this e-book—to help leaders like you anticipate challenges and seize opportunities in fintech and beyond.

To learn more about our work and explore additional resources, visit: www.devsu.com

devsu

Book your 30-minute Modernization Strategy Session with Devsu's engineering leadership.

This is a conversation with engineers, not sales. The agenda is your actual codebase, your actual constraints, and an honest read on whether a bounded pilot makes sense for your institution.

Visit devsu.com